```python
from a0_items import *

################################################################################
##########
### plugins - folders & sqlite file
################################################################################
###########

def create_folder (folder_path):

    if not os.path.isdir(folder_path):

        os.makedirs(folder_path)#if folder not exists, create it
        print (folder_path + ' folder NOT exists, is created')

    else:

        #shutil.rmtree(folder_path)#if folder exists, remove it
        print (folder_path + ' folder exists')


#create_folder (folder_path)


################################################################################
###############################
################################################################################
###############################


def get_ALL_in_folder (folder_path):

    LIST_items_in_folder = os.listdir(folder_path)

    return LIST_items_in_folder


#print(get_ALL_in_folder (folder_path ='../../data/raw/output_folder/'))

#print (len(get_ALL_in_folder (folder_path=('../../data/cooked/DB_squ/'))))



def get_ALL_subfolders_files_in_parent_folder (parent_folder_path):

    LIST_file = []
    LIST_folder =[]

    for root, dirs, files in os.walk(parent_folder_path, topdown=False):

        for name in files: # list files
            LIST_file.append(name)
            print(os.path.join(root, name))

        for name in dirs: #list folders
            LIST_folder.append(name)
            print(os.path.join(root, name))

    print ('Total files =>', len(LIST_file))
    print ('Total folders =>', len(LIST_folder))

    return LIST_folder, LIST_file


#get_ALL_subfolders_files_in_parent_folder (parent_folder_path= '../core_TASK/')
```

```
64
65
66
67
68    def remove_ALL_folder_items (folder_path):
69
70        LIST_folder_item = os.listdir (folder_path )
71
72        for x in LIST_folder_item:
73
74            ab_path_folder = os.path.abspath (folder_path + x) # Must convert relative path to
              absolute path
75
76            shutil.rmtree (ab_path_folder, ignore_errors=False)
77
78            print (ab_path_folder + ' => removed')
79
80
81    #remove_ALL_folder_items (folder_path)
82
83
84
85    def copy_folderA_to_folderB (folderA_path, folderB_path):
86
87        #Need to remove all in Folder B before copy Folder A to Folder B
88        remove_ALL_folder_items (folder_path = folderB_path)
89
90        #print (os.listdir(folder_TFM_SOURCE))
91
92        for y in LIST_folderA_item:
93
94            try:
95                ab_path_folderA = os.path.abspath (folderA_path + y) # Must convert relative
                  path to absolute path
96                ab_path_folderB = os.path.abspath (folderB_path + y) # Must convert relative
                  path to absolute path
97
98                copytree(ab_path_folderA, ab_path_folderB )
99                print (ab_path_folderB + '/' + ' => copied')
100
101           except Error as err:
102
103               print ('ERROR MSG => ', err)
104
105
106   #copy_folderA_to_folderB (folderA_path, folderB_path)
```